2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Online least-squares estimation of time varying systems with sparse temporal evolution and application to traffic estimation

A. Hofleitner, L. El Ghaoui and A. Bayen

***Abstract*— Using least-squares with an l_1-norm penalty is well-known to encourage sparse solutions. In this article, we propose an algorithm that performs online least-squares estimation of a time varying system with a l_1-norm penalty on the variations of the state estimate, leading to state estimates that exhibit few "jumps" over time. The algorithm analytically computes a path to update the state estimate as a new observation becomes available. The algorithm performs computationally efficient and numerically robust state estimation for time varying systems in which the dynamics are slow compared to the sampling rate.**

We use the algorithm for arterial traffic estimation with streaming probe vehicle data provided by the *Mobile Millennium* system and show a significant improvement in the estimation capabilities compared to a baseline model of traffic estimation. The estimation framework filters out the inherent noise of traffic dynamics and improves the interpretability and accuracy of the results. Results from an implementation in San Francisco on a network of more than 800 links using a fleet of 500 taxis sending their location every minute illustrate the possibility to use the algorithm to solve important practical estimation problems.

I. Problem statement and related work

L_1-norm in least-squares regression has attracted a lot of interest in the statistics [1], [2], signal processing [3], [4] and machine learning [5], [6] communities, in particular for estimation problems. Indeed, adding a l_1-penalty on the parameter vector leads to sparse solutions, which is a desirable property in order to achieve model selection [7], data compression [4], or for obtaining interpretable results. In this article, we present a way to use l_1-norm regularization to perform estimation of a time varying system. We assume that we receive sequential observations of the state of the system. As we receive a new observation, we update the estimate of the state online and we would like the *variations* in the estimates to be sparse.

We are sequentially given a set of training examples or observations $(y_i, a_i) \in \mathbb{R} \times \mathbb{R}^m, i = 1 \ldots n$. We wish to fit a linear model to estimate the response y_i as a function of the state vector $x \in \mathbb{R}^m$, $y_i = a_i^T x + \nu_i$, where ν_i represents the noise in the observation.

Least square optimization with a penalty on the l_1-norm of the parameter is known as the Lasso algorithm [1] and the resulting optimization problem is given by

$$x = \arg\min_{x \in \mathbb{R}^m} \frac{1}{2} \sum_{i=1}^{n} (a_i^T x - y_i)^2 + \mu_n ||x||_1 \qquad (1)$$

Ph.D. student, Electrical Engineering and Computer Science, UC Berkeley, CA (e-mail: aude@eecs.berkeley.edu). Corresponding author.
Professor, Electrical Engineering and Computer Science, UC Berkeley
Professor, Electrical Engineering and Computer Science and Civil and Environmental Engineering, UC Berkeley

where μ_n is a regularization parameter. The solution of (1) is typically sparse, i.e. the solution has few entries that are non-zero, and therefore identifies which dimensions in a_i are useful to predict y_i. In model selection for example, the different elements of a_i represent different features and the l_1 regularization selects the most relevant features to estimate y_i. There is no analytic formula for the optimal solution to the l_1-regularized least square. It is convex but not differentiable and requires specific algorithms to be solved efficiently at a large scale. It can be formulated as a convex *quadratic problem* (QP) with linear equality constraints and solved using standard interior-point methods which can handle medium-size problems [8]. A specialized interior-point method for large-scale problems was introduced in [9]. Other methods to solve (1) include iterative thresholding algorithms [10], [11], [12], feature-sign search [13], bound optimization methods [14] and gradient projection algorithms [15]. Homotopy methods have also been applied to the Lasso to compute the full regularization path when the regularizaion parameter μ_n varies [16], [17], [18]. They are particularly efficient when the solution is very sparse [19]. When the training examples $(y_i, a_i)_{i=1\ldots n}$ are obtained sequentially, Garrigues et al [20] present a homotopy algorithm to compute the solution of the Lasso problem after receiving n observations from the solution of the Lasso after receiving $n-1$ observations. This method is particularly efficient when the supports of the two solutions are close. Note that to address the issue of the non-smoothness of the l_1 norm, most lasso algorithms optimize the dual of (1). For these algorithms, the solution computed with $n-1$ observations may not be used as a warm start to compute the solution with n observations as it may no longer be feasible as we add new observations.

In this article, the vector x^n is the state estimate of the system after receiving the n^{th} observation. We are interested in sparse changes in the state vector as we receive new observations. To achieve this property, we add a l_1 penalty on the variation of the state vector, which regularizes the estimates when measurements are noisy and the dynamics of the system are slow compared to the sampling rate. We would like to re-solve the problem as we get a new observation using the information already available and without having to *completely* resolve the problem. This is akin to recursive least-squares, but now we have to handle the l_1-norm term added to the objective function. The estimation problem of x^n is defined recursively as:

$$x^n = \arg\min_{x \in \mathbb{R}^m} \frac{1}{2} \sum_{i=1}^{n} (a_i^T x - y_i)^2 + \mu_n ||x - x^{n-1}||_1. \qquad (2)$$

The algorithm is initialized assuming information x_{init} on the state of the system (*e.g.* historical or previous estimate) and we set $x^0 = x_{\text{init}}$. When estimating x^n, we want the vector $x^n - x^{n-1}$ to be sparse. We call x^{n-1} the *reference parameter* for the estimation of x_n. After receiving a new observation (y_{n+1}, a_{n+1}), we define a homotopy algorithm to update the estimate from x^n to x^{n+1} and vary the reference parameter from x^{n-1} to x^n, before receiving the next data point.

In applications, it is useful to add additional regularization to the optimization problem (2). In particular, if we call A the matrix whose i^{th} row is equal to a_i^T, the matrix $A^T A$ should be non singular for the solution of the least-squares estimation problem to be unique. Moreover, the regularization term $\mu_n ||x - x^{n-1}||_1$ is on the difference in successive parameters but there is no regularization to maintain the state estimates within given bounds (corresponding to physical characteristics of the system for example) or close to an apriori estimate of the state. We show how to leverage prior information $\hat{x}$ on the value of the parameter x (from historical data for example) by adding a l_2 regularization term to problem (2), with weighting parameter λ. We also propose an algorithm that ensures that the estimates remain within given upper and lower bounds $\bar{x}$ and $\underline{x}$. The resulting estimation scheme amounts to solving the following optimization problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^m}{\textbf{minimize}} \; & \frac{1}{2} \sum_{i=1}^{n} (a_i^T x - y_i)^2 + \mu_n ||x - x^{n-1}||_1 + \frac{\lambda}{2} ||x - \hat{x}||_2^2 \\ \textbf{s.t.} \; & \underline{x} \leq x \leq \bar{x} \end{aligned} \tag{3}$$

where the inequalities $\underline{x} \leq x$ and $x \leq \bar{x}$ refer to componentwise inequalities on the vectors. This article presents an online estimation algorithm of a time varying system with sparse changes on the state vector between successive estimations. The algorithm is based on homotopy algorithms for the variation of the regularization parameter μ_n [16], [17], [18] and the addition of a new observation as it becomes available [20]. The contribution of this article is the presentation of a homotopy algorithm to produce sparse variations in the state estimate and in particular, update the reference parameter each time a new observation is received. This algorithm is particularly efficient to solve large scale online estimation problems, as demonstrated in this article. We present the estimation algorithm with an additional l_2-regularization and bounds on the state vector, necessary to regularize the state estimates and ensure that the results meet physical requirements of the system.

This article is organized as follows. We review the optimality conditions of the Lasso algorithm (Section II) and present an algorithm to compute a homotopy on the parameter vector x solving the optimization problem (3) (Section III). We apply these results to traffic estimation on an arterial network in San Francisco, CA (Section IV). We estimate the mean travel time on each link of the network, totalling more than 12.6 kilometers of roadway, from GPS data sent by 500 probe vehicles sampled every minute (Figure 1). The data is collected by the *Mobile Millennium* system [21] which receives on the order of 500,000 points daily in the San Francisco Bay Area. We discuss possible extensions of this work in Section V.

Fig. 1. San Francisco taxi measurement locations, observed at a rate of once per minute. Each black dot represents the measurement of the location of a taxi, received between midnight and 7am, on March 29th, 2010. The large red dots represent the location of taxis at 7am on that day.

II. OPTIMALITY CONDITIONS FOR THE LASSO

The objective function in (1) is convex and non-smooth since the l_1-norm is not differentiable when there exists i such that the i^{th} element of x (denoted x_i) equals zero. Hence there is a global minimum at x if and only if the subdifferential of the objective function at x contains the 0-vector. The subdifferential of the l_1-norm at x is the following set

$$\partial \|x\|_1 = \left\{ v \in \mathbb{R}^m : \left\{ \begin{array}{lllll} v_i & = & \text{sgn}(x_i) & \text{if} & |x_i| > 0 \\ v_i & \in & [-1, 1] & \text{if} & x_i = 0 \end{array} \right\} \right.$$

where $\text{sgn}(\cdot)$ is the sign function. Let $A \in \mathbb{R}^{n \times m}$ be the matrix whose i^{th} row is equal to a_i^T, and let $y = (y_1, \ldots, y_n)^T$ be the vector of response variables. The optimality conditions for the Lasso problem (1) are given by $A^T(Ax - y) + \lambda v = 0$, $v \in \partial \|x\|_1$.

We define as the *active set* (resp. *non active set*), the indices representing non-zero elements (resp. zero elements) of x. The active and non-active sets are referenced by the subscripts 1 and 2 respectively. For example A_1 is the matrix representing the columns of A in the active set. The vector x_1 represents the non-zero coordinates of x and x_2 is the 0-vector. The index 1_i (resp. 2_i) references the i^{th} coordinate of the active (resp. non active set). Since $v \in \partial \|x\|_1$, the i^{th} coordinate of v_1 is $v_{1_i} = \text{sgn}(x_{1_i})$, and the i^{th} coordinate of v_2 is $v_{2_i} \in [-1, 1]$. If the solution is unique, it can be shown that $A_1^T A_1$ is invertible, and we can rewrite the optimality conditions as

$$\begin{array}{rcl} x_1^* & = & (A_1^T A_1)^{-1} (A_1^T y - \mu_n v_1) \\ -\mu_n v_2 & = & A_2^T (A_1 x_1^* - y) \end{array}$$

Note that if we know the active set and the signs of the coefficients of the solution, thus the vector v_1, we can compute it in closed form. When solving the estimation problem (2) or (3), we define a change of variable $x_r = \varphi(x)$ such that if x is the solution of the estimation problem, x_r is typically sparse. We present an algorithm to update the active set and the signs of the coefficients of the solution

x_r from previous solutions by computing a continuous path between the successive state estimates.

III. Homotopy algorithm

Suppose that we have computed the solution x^n to Equation (3) with n observations. The reference parameter is x^{n-1} since we would like the variations between x^{n-1} and x^n to be sparse. We receive an additional observation $(y_{n+1}, a_{n+1}) \in \mathbb{R} \times \mathbb{R}^m$ and a new penalty coefficient μ_{n+1}. In general, μ_n reflects the number of measurements. We also need to update the reference parameter from x^{n-1} to x^n. We introduce the following optimization problem:

$$x(t,u,\mu) = \arg\min \left\{ \frac{1}{2} \left\| \begin{pmatrix} A \\ t a_{n+1}^T \end{pmatrix} x - \begin{pmatrix} y \\ t y_{n+1} \end{pmatrix} \right\|_2^2 + \mu \left\| x - \left((1-u) x^{n-1} + u x^n \right) \right\|_1 + \frac{\lambda}{2} ||x - \hat{x}||_2^2 \right\}.$$

where the minimum is taken over the set of $x \in \mathbb{R}^m$ such that $\underline{x} \leq x \leq \bar{x}$ and we assume that $\hat{x}$ is within the bounds $\underline{x}$ and $\bar{x}$. We have $x(0,0,\mu_n) = x^n$ and $x(1,1,\mu_{n+1}) = x^{n+1}$. We propose an algorithm that computes a path from x^n to x^{n+1} in three steps (Algorithm 1):

- **Step** 1: *Add an observation* (Section III-A): vary t from 0 to 1 with $\mu = \mu_n$ and $u = 0$.
- **Step** 2: *Vary the regularization* (Section III-B): vary μ from μ_n to μ_{n+1} with $t = 1$ and $u = 0$.
- **Step** 3: *Update the reference parameter* from x^{n-1} to x^n (Section III-C): vary u from 0 to 1 with $\mu = \mu_{n+1}$ and $t = 1$.

A. Step 1: adding a new data point

We introduce a change of variables to formulate the optimization problem as a Lasso problem and solve it using an online Lasso algorithm [20]. We define $x_r = x - x^{n-1}$ and solve the optimization problem for x_r. The vector x_r must satisfy coordinate by coordinate inequalities $\underline{x}_r \leq x_r \leq \bar{x}_r$, with $\underline{x}_r = \underline{x} - x^{n-1}$ and $\bar{x}_r = \bar{x} - x^{n-1}$. If the i^{th} element of x_r equals $\bar{x}_{r,i}$ (resp. $\underline{x}_{r,i}$), we say that the upper (resp. lower) bound i is an *active constraint* and reference the set of active constraints by the subscript c. We reference the i^{th} coordinate of the active constraints by the index c_i and define $\varepsilon_{c_i} = -1$ (resp. $\varepsilon_{c_i} = 1$) if the upper (resp. lower) bound is active. We denote by E_c the diagonal matrix with i^{th} diagonal element equal to ε_{c_i}.

The matrix A and the vector y represent $n+1$ observations of the state. Given the set of active constraints c, we define $y_r = y - Ax^{n-1} - A_c x_{r,c}$. We define $b_j = A_j^T y_r + \lambda [\hat{x} - x^{n-1}]_j$, where j represents the set of indices 1 or 2 (active and non-active). We define $b_c = A_c^T y_r + \lambda [\hat{x} - x^{n-1}]_c - \lambda x_{r,c}$ and $K = (A_1^T A_1 + \lambda I)^{-1}$. We note a_{n+1}^T the last row of A and $y_{r,n+1}$ the last element of y_r, which correspond to the last observation received. Let $x_r^*(t)$ be the solution of the following optimization problem:

$$\begin{aligned} \min_{x_r \in \mathbb{R}^m} \; & \tfrac{1}{2} ||A x_r - y_r - A_c x_{r,c}||_2^2 + \tfrac{\lambda}{2} ||x_r - (\hat{x} - x^{n-1})||_2^2 \\ & + \tfrac{1-t^2}{2} ||a_{n+1}^T x_r - y_{r,n+1} - a_{n+1,c}^T x_{r,c}||_2^2 + \mu_n ||x_r||_1 \\ \textbf{s.t.} \; & \underline{x}_r \leq x_r \leq \bar{x}_r \end{aligned}$$

We write the optimality conditions for each set of indices (active indices, active constraints, non active indices):

$$\begin{aligned} (A_1^T A_1 + \lambda I) x_{r,1}^*(t) - b_1 + (t^2-1) a_{n+1,1} (a_{n+1,1}^T x_{r,1}^*(t) - y_{r,n+1}) + \mu v_1 &= 0 \\ E_c \left(A_c^T A_1 x_{r,1}^*(t) - b_c + (t^2-1) a_{n+1,c} (a_{n+1,1}^T x_{r,1}^*(t) - y_{r,n+1}) + \mu v_c \right) &\geq 0 \\ A_2^T A_1 x_{r,1}^*(t) - b_2 + (t^2-1) a_{n+1,2} (a_{n+1,1}^T x_{r,1}^*(t) - y_{r,n+1}) + \mu w_2(t) &= 0 \end{aligned}$$

With this change of variable, the derivation of $x_{r,1}^*(t)$ is similar to the calculation performed in [20]. We use the Sherman-Morrison formula and solve for $x_{r,1}^*(t)$:

$$x_{r,1}^*(t) = \tilde{x}_{r,1} - \frac{(t^2-1)\bar{e}}{1 + \alpha(t^2-1)} q$$

with $\tilde{x}_{r,1} = K(b_1 - \mu v_1)$, $\bar{e} = a_{n+1,1}^T \tilde{x}_{r,1} - y_{r,n+1}$, $q = K a_{n+1,1}$ and $\alpha = a_{n+1,1}^T K a_{n+1,1}$. We denote by t_{1_i}, the value of t that sets the i^{th} coordinate of $x_{r,1}^*(t)$ to zero.

$$t_{1_i} = \left(1 + \left(\frac{\bar{e} q_i}{\tilde{x}_{r,1_i}} - \alpha \right)^{-1} \right)^{1/2}$$

We denote by $t_{\bar{c}_i}$ (resp. $t_{\underline{c}_i}$) , the value of t that sets the parameter $x_{r,1_i}^*$ to $\bar{x}_{r,1_i}$ (resp. $\underline{x}_{r,1_i}$).

$$t_{\bar{c}_i} = \left(1 + \left(\frac{\bar{e} q_i}{\tilde{x}_{r,1_i} - \bar{x}_{r,1_i}} - \alpha \right)^{-1} \right)^{1/2}$$

and have a same expression for $t_{\underline{c}_i}$, replacing $\bar{x}_r$ by $\underline{x}_r$. We notice that

$$\begin{cases} a_{n+1,1}^T x_{r,1}^*(t) - y_{r,n+1} &= \frac{\bar{e}}{1+\alpha(t^2-1)} \\ A_1 x_{r,1}^*(t) - y_r &= \tilde{e} - \frac{(t^2-1)\bar{e}}{1+\alpha(t^2-1)} A_1 q \end{cases}$$

where $\tilde{e} = A_1 \tilde{x}_{r,1} - y_r$. We replace these expressions in the optimality conditions for the active constraint indices c and the non active indices 2. Let t_{c_i} be the value of t such that the i^{th} element of $x_{r,c}$ leaves the set of active constraints c, as the partial derivative of the objective function equals zero:

$$A_{c_i}^T \tilde{e} + \mu v_{c_i} + \lambda (x_{c_i} - [\hat{x} - x^{n-1}]_{c_i}) + \frac{t_{c_i}^2 - 1}{1 + \alpha(t_{c_i}^2 - 1)} \left(a_{n+1,c_i} - A_{c_i}^T A_1 q \right) \bar{e} = 0,$$

$$\text{so } t_{c_i} = \left(1 + \left(\frac{\bar{e}(a_{n+1,c_i} - A_{c_i}^T A_1 q)}{-\mu v_{c_i} - \lambda (x_{r,c_i} - [\hat{x} - x^{n-1}]_{c_i}) - A_{c_i}^T \tilde{e}} - \alpha \right)^{-1} \right)^{1/2}.$$

Let $t_{2_i}^+$ (resp. $t_{2_i}^-$) be the value such that $w_{2_i}(t_{2_i}^+) = 1$ (resp. $w_{2_i}(t_{2_i}^-) = -1$). We have

$$\begin{aligned} t_{2_i}^+ &= \left[1 + \left(-\alpha + \frac{\bar{e}(a_{n+1,2_j} - A_{2_i}^T A_1 q)}{-\mu + \lambda (\hat{x} - x^{n-1})_{2_i} - A_{2_i}^T \tilde{e}} \right)^{-1} \right]^{1/2} \\ t_{2-j} &= \left[1 + \left(-\alpha + \frac{\bar{e}(a_{n+1,2_j} - A_{2_i}^T A_1 q)}{+\mu + \lambda (\hat{x} - x^{n-1})_{2_i} - A_{2_i}^T \tilde{e}} \right)^{-1} \right]^{1/2} \end{aligned}$$

These derivations allow the computation of a continuous update of the state as a new observation is received.

B. Step 2: varying the regularization parameter

We use the change of variables from Section III-A and solve the Lasso problem where we vary the penalization parameter [16], [17], [18]. With a slight abuse of notation on arguments (but for notational simplicity), we denote by $x_r^*(\mu)$ the solution of the optimization problem

$$\begin{aligned}\min_{x_r\in\mathbb{R}^m} \cdot\ & \frac{1}{2}||Ax_r - y_r - A_c x_{r,c}||_2^2 + \mu||x_r||_1 + \frac{\lambda}{2}||x_r - (\hat{x} - x^{n-1})||_2^2. \\ \text{s.t.}\ \ & \underline{x}_r \le x_r \le \bar{x}_r\end{aligned}$$

The information on the active set and the signs of the parameters in the active set is available at $\mu = \mu_n$ (Step 1). The active set, active constraints and signs remain constant for μ in an interval $[\mu_n, \mu^*)$ where the solution $x_r^*(\mu)$ is affine in μ. As we reach the "transition point" μ^*, we update the active set, active constraints and signs which remain valid until the next transition point. The optimality conditions read:

$$\begin{aligned}(A_1^T A_1 + \lambda I)x_{r,1}^*(\mu) - b_1 + \mu v_1 = 0 \\ E_c\left(A_c^T A_1 x_{r,1}^*(\mu) - b_c + \mu v_c\right) \succeq 0 \\ A_2^T A_1 x_{r,1}^*(\mu) - b_2 + \mu w_2(\mu) = 0\end{aligned}$$

Solving for $x_{r,1}^*(\mu)$, we have $x_{r,1}^*(\mu) = K(b_1 - \mu v_1)$. Denoting by μ_{1_i}, the value of μ that sets the i^{th} coordinate of $x_{r,1}^*$ to zero, we read $\mu_{1_i} = [Kb_1]_i / [Kv_1]_i$.
The upper bound (resp. lower bound) becomes active as $x_{r,1_i}^*(\mu)$ equals $\bar{x}_{r,1_i}$ (resp. $\underline{x}_{r,1_i}$), when μ equals $\mu_{\bar{c}_i}$ (resp. $\mu_{\underline{c}_i}$), *i.e.* $\mu_{\bar{c}_i} = [Kb_1 - \bar{x}_r]_i / [Kv_1]_i$.

Let μ_{c_i} be the value of μ such that the i^{th} component of $x_{r,c}$ is no longer an active constraint. Let $\mu_{2_i}^+$ (resp. $\mu_{2_i}^-$) be the value of μ such that $w_{2_i}(\mu) = 1$ (resp. $w_{2_i}(\mu) = -1$). From the expression of $x_{r,1}^*(\mu)$ and the optimality conditions, we see that the partial derivatives of the objective function for coordinates in c and the function $\mu \mapsto w_2(\mu)$ are affine in μ. We derive the expressions of μ_{c_i}, $\mu_{2_i}^+$ and $\mu_{2_i}^-$ as

$$\begin{aligned}\mu_{c_i} &= (b_{c_i} - A_{c_i}^T A_1 K b_1)/(v_c - A_{c_i}^T A_1 K v_1) \\ \mu_{2_i}^+ &= (b_{2_i} - A_{2_i}^T A_1 K b_1)/(1 - A_{2_i}^T A_1 K v_1) \\ \mu_{2_i}^- &= (b_{2_i} - A_{2_i}^T A_1 K b_1)/(-1 - A_{2_i}^T A_1 K v_1)\end{aligned}$$

C. Step 3: Updating the reference parameter

After adding the new observation and varying the regularization parameter, the algorithm updates the reference parameter from x^{n-1} to x^n. We define $\Delta x = x^n - x^{n-1}$ and the change of variable $x_r(u) = x - [(1-u)x^{n-1} + ux^n]$, which is the vector that we impose sparsity on. The constraints on this variable are $\underline{x}_r(u) = \underline{x} - [(1-u)x^{n-1} + ux^n]$ and $\bar{x}_r(u)$ (defined similarly from $\bar{x}$). Given a value of u and a set of active constraints c, we define $x_{r,c}^0 = x_{r,c}(u) + u(\Delta x)_c$ and notice that this vector no longer depends on u. The i^{th} element of this vector is equal to $\underline{x}_{c_i} - x_{c_i}^{n-1}$ (resp. $\bar{x}_{c_i} - x_{c_i}^{n-1}$) if the lower (resp. upper) bound is active. We define $y_r = y - Ax^{n-1} - A_c x_{r,c}^0$. We

---

Algorithm 1 Homotopy algorithm for online state estimation of time varying systems with sparse temporal changes.

1. *Add the latest observation* (a_{n+1}, y_{n+1})*:* Compute the path from $x^n = x(0,0,\mu_n)$ to $x(1,0,\mu_n)$. Refer to the derivations of this article and to [16], [17], [18] for the details of the algorithm.
2. *Vary the regularization parameter:* Compute the path from $x(1,0,\mu_n)$ to $x(1,0,\mu_{n+1})$. Refer to the derivations of the article and to [20] for the details of the algorithm.
3. Initialize the active set to the non-zero coefficients of $x_r(0) = x(1,0,\mu_{n+1}) - x^{n-1}$. Initialize $v_1 = \text{sgn}(x_{r,1}^*(0))$, $u^* = 0$ and $K = (A_1^T A_1 + \lambda I)^{-1}$.
4. Compute the next transition point u^*. If it is smaller than the previous transition point or greater than 1, go to Step 6. Otherwise:
   a. The i^{th} component of $x_{r,1}^*(u^*)$ goes to zero: remove i from the active set.
   b. The i^{th} component of $x_{r,1}^*(u^*)$ reaches $\underline{x}_r(u)$ or $\bar{x}_r(u)$: remove i from the active set and add it to the active constraints.
   c. The i^{th} component of $w_2(u^*)$ reaches one in absolute value: add i to the active set. If the component reaches 1 (resp. -1), then set $v_i = 1$ (resp. $v_i = -1$).
   d. The i^{th} optimality condition of the active constraints reaches zero: add i to the active set.
5. Update v_1 and A_1 according to the updated active set and sign of the parameters. Update $K = (A_1^T A_1)^{-1}$, (rank 1 update). Go to Step 4.
6. Compute the final value of $x_r^*(1)$.

---

keep the notation $b_j = A_j^T y_r + \lambda[\hat{x} - x^{n-1}]_j$, $j \in \{1,2\}$, and define $b_c = A_c^T y_r + \lambda[\hat{x} - x^{n-1}]_c - \lambda x_{r,c}^0$. With this notation, $x_r^*(u)$ is the minimizer of the optimization problem

$$\begin{aligned}\min_{x_r\in\mathbb{R}^m} \cdot\ & \frac{1}{2}||Ax_r - y_r + uA\Delta x - A_c x_{r,c}^0||_2^2 + \mu_{n+1}||x_r||_1 \\ & + \frac{\lambda}{2}||x_r - (\hat{x} - x^{n-1}) + u\Delta x||_2^2 \\ \text{s.t.}\ \ & \underline{x}_r(u) \le x_r \le \bar{x}_r(u)\end{aligned}$$

The solution at $u = 0$ is $x(1,0,\mu_{n+1})$, computed by Step 2. The optimality conditions read

$$\begin{aligned}(A_1^T A_1 + \lambda I)x_{r,1}^*(u) - b_1 + \mu v_1 + u\left(A_1^T(A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_1\right) = 0 \\ E_c\left(A_c^T A_1 x_{r,1}^*(u) - b_c + \mu v_c + u\left(A_c^T(A\Delta x - A_c(\Delta x)_c)\right)\right) \ge 0 \\ A_2^T A_1 x_{r,1}^*(u) - b_2 + \mu w_2(u) + u\left(A_2^T(A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_2\right) = 0\end{aligned}$$

The solution $x_{r,1}^*(u)$ is affine in u: $x_{r,1}^*(u) = \xi + u\chi$, with $\xi = K(b_1 - \mu v_1)$ and $\chi = -K\left(A_1^T(A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_1\right)$.

We solve for u_{1_i} that sets the i^{th} component of $x_{r,1}^*$ to zero, and have $u_{1_i} = -\xi_i/\chi_i$. The i^{th} component of the active set reaches the lower bound $\underline{x}_{r,1_i}(u)$ for $u = u_{\underline{c}_i}$. Note that $\underline{x}_{r,1_i}(u) = \underline{x}_{r,1_i}(0) - u(\Delta x)_{1_i}$, so $u_{\underline{c}_i} = (\underline{x}_{r,1_i}(0) - \xi_i)/(\chi_i + (\Delta x)_{1_i})$, and we derive a similar expression for $u_{\bar{c}_i}$.

We also solve for u_{c_i} such that the i^{th} coefficient of $x_{r,c}(u)$ is no longer an active constraint, and thus the partial

Fig. 2. Subnetwork of San Francisco used for arterial traffic estimation.

derivative of the objective function equals zero:

$$A_{c_i}^T A_1 \xi - b_{c_i} + \mu v_{c_i} + u_{c_i} \left(A_{c_i}^T (A_1 \chi + A\Delta x - A_c(\Delta x)_c) \right) = 0,$$

from which we read the expression of u_{c_i}.

We solve for $u_{2_i}^+$ (resp. $u_{2_i}^-$), where $u_{2_i}^+$ (resp. $u_{2_i}^-$) is the value of u for which the i^{th} component of $x_{r,2}$ enters the active set and becomes positive (resp. negative). They are given by:

$$u_{2_i}^+ = -\frac{A_{2_i}^T A_1 \xi - b_{2_i} + \mu}{A_{2_i}^T (A_1 \chi + A\Delta x - A_c(\Delta x)_c) + \lambda (\Delta x)_{2_i}}$$

$$u_{2_i}^- = -\frac{A_{2_i}^T A_1 \xi - b_{2_i} - \mu}{A_{2_i}^T (A_1 \chi + A\Delta x - A_c(\Delta x)_c) + \lambda (\Delta x)_{2_i}}$$

IV. Experimental results

We apply the algorithm to arterial traffic estimation on a subnetwork of San Francisco, CA consisting in 815 links (Figure 2). Arterial traffic is modeled as a time varying system and we seek to estimate travel times on each link of the network, as they vary over time. Traffic data on arterial networks is mainly provided from probes sending their location at a given sampling frequency (common sampling frequencies are around 1 minute). The proportion of sampled vehicles (penetration rate) remains limited and rarely exceeds a few percent of the vehicles traveling on the network. Moreover, traffic signals cause important variation on the travel time experienced on a link of the network within very short periods of time (depending on whether the vehicle stopped at the signal or not), while the actual changes in traffic conditions have slower dynamics. Given the penetration rate of probe vehicles, we seek to estimate trends in traffic conditions rather than fluctuations around a mean value. For these reasons, arterial traffic estimation is a good application for the algorithm. The parameter x^n represents the average travel time on each link after receiving the n^{th} observation and we impose sparsity on its temporal evolution.

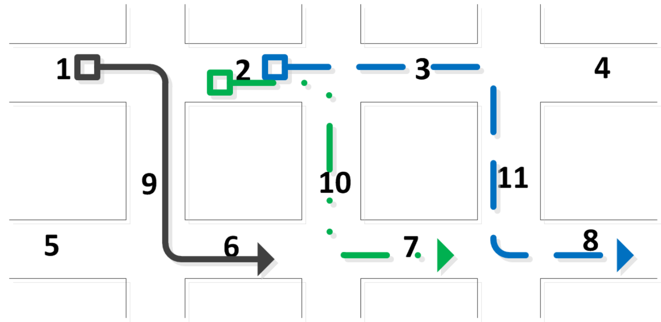

Fig. 3. Paths of three probe vehicles on a network with eleven links. The path of a probe is represented as a vector $a_i \in [0,1]^{11}$ where the j^{th} coordinate of a_i represent the fraction of link j traveled by the probe. The path represented with a solid line is represented with a sparse vector with non zero coordinates 1, 6 and 9, respectively equal to 0.4, 0.7 and 1 considering that the probe traveled 40% of link 1 and 70% of link 6. The vector representing the dashed path has non zero coordinates 2, 3, 8 and 11, respectively equal to 0.3, 1, 0.8 and 1 considering that the probe traveled 30% of link 2 and 80% of link 8.

Experimental setup: Beginning in March of 2009, data has been collected from probe vehicles in the San Francisco Bay Area by the *Mobile Millennium* system [21]. A fleet of over 500 taxis report their location every minute, along with an identifier and a status (carrying a passenger or not) [22] allowing filtering of the taxi stops to load or unload passengers. The duration between two successive location reports z_1 and z_2 represents an observation of the travel time y_i of the vehicle on its path from z_1 to z_2. We use an algorithm [23] that combines models of GPS measurements and drivers' behavior into a conditional random field to provide trajectory reconstructions between z_1 and z_2. The latency in the communication of the location data to our servers is generally less than a few minutes.

Each trajectory (path) is converted in a vector $a_i \in [0, 1]^m$, where m is the number of links in the arterial network. The k^{th} coordinate of a_i, $a_{i,k}$ is the fraction of the link traveled by the probe vehicle, computed as the distance traveled on the link divided by the length of the link. In particular, $a_{i,k} = 0$ if the vehicle did not travel on link k and $a_{i,k} = 1$ if the vehicle fully traversed the link k (see Figure 3). Note that on arterial links the mean travel time on a fraction of the link does not vary proportionally with the distance traveled. Vehicles are more likely to experience delays close to the downstream intersection because of the presence of traffic signals [24], [25], and thus the coefficients $a_{i,k}$ should take into account the locations where the vehicles started and ended their travel on link k. However, these considerations are not taken into account in this article.

Numerical experiments: We learn the parameter x, solving equation (3), which represents the mean travel time on each link of the network. We initialize the algorithm using a previous estimate of the mean travel times given by least-squares regression and use historical mean travel times for the l_2 regularization $\hat{x}$. Each time we receive a new travel time observation, we add the new observation $(y_{n+1}, a_{n+1}) \in \mathbb{R} \times \mathbb{R}^m$ (Section III-A), increase the regularization parameter from $n\mu$ to $(n+1)\mu$ (Section III-B) and

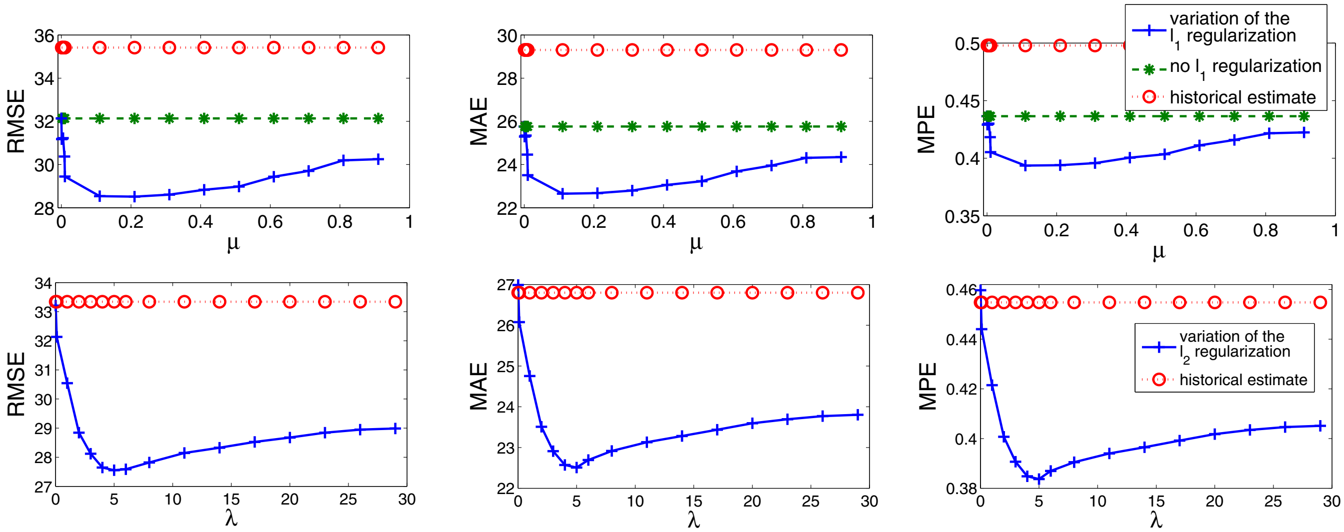

Fig. 4. Variation of the error metrics depending on the regularization. Both the l_1 and l_2 regularizations improve the estimation accuracy and the regularization parameters can be chosen optimally. The three top figures represent the effect of the l_1 regularization for the estimation accuracy. The three bottom figures show the importance of the additional l_2 regularization introduced in Section III for the robustness of the estimation.

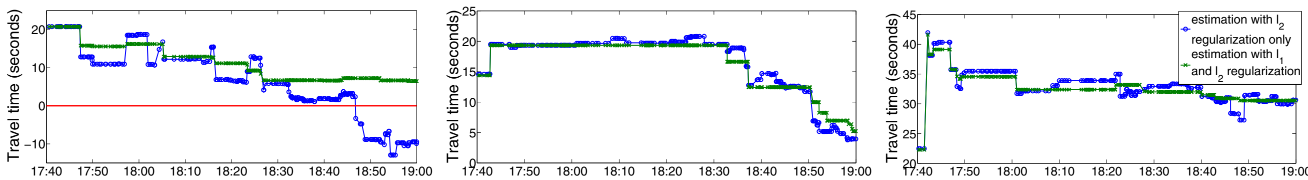

Fig. 5. Qualitative evolution of the travel time estimates on different links of the network. The l_1 regularization provides more stable estimates that represent the dynamics of traffic more accurately and increase the physical interpretation. The left figure shows that the estimation with l_2 regularization leads to estimates that are not physically possible (negative travel times), while the estimate with l_1 regularization remains within feasible bounds. On all figures, the l_2 estimate is noisy while the additional l_1 regularization remains constant between each temporal transitions in traffic conditions.

update the reference parameter (Section III-C). As we are interested in the estimation of the current state of the system, observations of the state may only be relevant for a *limited period of time*. We consider that each observation remain relevant for a time T after being received in the system. When an observation $(y_j, a_j) \in \mathbb{R} \times \mathbb{R}^m$ becomes obsolete, we (1) remove the old observation $(y_j, a_j) \in \mathbb{R} \times \mathbb{R}^m$ by decreasing t from 1 to 0, (2) decrease the regularization parameter from $(n+1)\mu$ to $n\mu$, where we assume that n represents the number of observations currently considered relevant for the estimation and (3) update the reference parameter.

We want to assess the performance of the model and quantify the effect of the regularization parameters λ and μ. The first parameter penalizes solutions which are far (in the l_2-norm sense) from the historical estimate of travel times $\hat{x}$. The second parameter imposes sparsity on the variation of the state. The choice of these parameters leads to a compromise between (1) fitting the data, with risks of overfitting and lack of physical interpretation and (2) putting too much weight on the regularization and not estimating accurately the current state of the system.

In this case study, we estimate traffic conditions using taxi data collected on September 3, 2010 from 5:00pm to 7:00pm in a subnetwork of San Francisco. This subnetwork contains 815 links (where a link is defined as the road between two signals) totalling more than 12.6 kilometers of roadway.

We use cross-validation to assess the performance of our model, randomly splitting the observations sent by the probe vehicles between a training set and a validation set. After learning the travel time estimates on the training set, we use the validation set to compare our estimates to the travel time observations. We compare the performance of our model with a *baseline model*, which uses the historical value of the link travel times $\hat{x}$ as the estimate of the state. We consider three validation metrics which, even though closely related, give different information on the quality of the estimation: the root mean squared error (RMSE), the mean absolute error (MAE) and the mean percentage error (MPE)[1]. The algorithm minimizes the l_2 norm of the residual between the estimate and the observation. The RMSE indicates the goodness of fit with validation data. Note that the variability of arterial travel times (due to traffic signals, pedestians, etc.) leads to important fluctuations of travel times. This inherent variability in the state of the system makes our estimation model robust with sparse variations, but is also responsible for relatively high values of the error metrics. For example, the RMSE is greater than the standard deviation of travel times [24].

The results indicate that both the l_1 and the l_2 regularization (Figure 4) are important to improve the estimation capabilities. For a wide range of parameters, the results are

[1] $\mathrm{RMSE} = \sqrt{\frac{\sum_{o=1}^{O}(y_o - \hat{y}_o)^2}{O}}$ $\quad \mathrm{MAE} = \frac{\sum_{o=1}^{O}|y_o - \hat{y}_o|}{O}$ $\quad \mathrm{MPE} = \frac{1}{O}\sum_{o=1}^{O}\frac{|y_o - \hat{y}_o|}{y_o}$

significantly better than the baseline model. The results also underline the importance of the additional l_2 regularization to improve the robustness of the algorithm. Figure 5 illustrates that in addition to improving the estimation capabilities, the algorithm produces results that are easier to interpret. Arterial traffic is highly variable and the variability often prevents the interpretation of the results. With this model, we are able to deal with this variability in order to estimate the trends in travel times on the links of the network.

V. Conclusion and possible extensions

This article derives an algorithm for an online least-squares estimation of the state x of a time varying system from successive observations $(y_i, a_i) \in \mathbb{R} \times \mathbb{R}^m$. We use l_1-norm regularization to limit the variations in the estimate of the state to capture the trends in the dynamics rather than the fluctuations. We add l_2-norm regularization to increase the robustness of the estimator and limit numerical issues when the matrix $A^T A$ is singular (or ill-conditioned), where A is the matrix with line i equal to a_i^T. Constraints ensure that the state estimates remain within feasible bounds.

The homotopy algorithm is particularly efficient when the variations between the estimates are sparse, leading to few transition points. The algorithm computes a continuous path, which is in general not possible for other lasso algorithms which solve the dual problem. Moreover, the computational costs are limited as all matrix inverses are computed with rank 1 updates. The number of transition points and active indices varies with the parameter μ. As μ increases, the number of transition points and active indices decreases, improving the computational efficiency of the algorithm. For small values of μ, the algorithm may not be as efficient, as the number of transitions is bounded by 3^m.

The model provides a significant improvement in the estimation capabilities, compared to a baseline model of arterial traffic estimation with probe data. We achieve sparse variations in the parameter and estimate the global trends in traffic conditions by filtering out the noise due to fluctuation. The number of transition points and active indices remain small throughout the algorithm (inferior to ten for a network with 815 links). This algorithm could be developed further to study change detection in time varying systems. We are also investigating generalization of this algorithm to more general forms of l_1-norm regularizations. For example, we could be interested in sparse spatial variations of the estimates.

Acknowledgments

The authors wish to thank Timothy Hunter and Ryan Herring from UC Berkeley for providing filtered probe trajectories from the raw measurements of the probe vehicles as well as valuable visualization capabilities of the data. We also thank Ryan Herring for his help in proofreading. We thank the members of the staff of the California Institute for Innovative Transportation for their contributions to develop, build, and deploy the system infrastructure of *Mobile Millennium* on which the numerical experiments of this article rely.

References

[1] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[2] Y. Dodge, *Statistical data analysis based on the L1-norm and related methods*. Birkhauser, 2002.

[3] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, p. 118, 2007.

[4] E. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, vol. 3, 2006, pp. 1433–1452.

[5] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[6] A. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[7] E. Candès and Y. Plan, "Near-ideal model selection by l1 minimization," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2145–2177, 2009.

[8] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[9] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale l1-Regularized Least Squares," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 606–617, 2008.

[10] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[11] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[12] C. Rozell, D. Johnson, R. Baraniuk, and B. Olshausen, "Locally competitive algorithms for sparse approximation," in *IEEE International Conference on Image Processing, ICIP 2007*, vol. 4, 2007.

[13] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," *Advances in neural information processing systems*, vol. 19, p. 801, 2007.

[14] M. Figueiredo and R. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *IEEE International Conference on Image Processing, ICIP 2005.*, vol. 2, 2005.

[15] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 586–597, 2008.

[16] M. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *IMA journal of numerical analysis*, vol. 20, no. 3, p. 389, 2000.

[17] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[18] D. Malioutov, M. Cetin, and A. Willsky, "Homotopy continuation for sparse signal representation," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing.*, Philadelphia, Pennsylvania, USA, 2005, pp. 733–736.

[19] I. Drori and D. Donoho, "Solution of l1 Minimization Problems by LARS/Homotopy Methods," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, vol. 3, 2006.

[20] P. Garrigues and L. El Ghaoui, "An homotopy algorithm for the Lasso with online observations," in *Neural Information Processing Systems (NIPS)*, vol. 21, 2008.

[21] "The *Mobile Millennium* Project," http://traffic.berkeley.edu.

[22] "Cabspotting," http://www.cabspotting.org.

[23] T. Hunter, T. Moldovan, M. Zaharia, J. Ma, S. Merzgui, M. Franklin, and A. Bayen, "Scaling the mobile millennium system in the cloud," *ACM Symposium on Cloud Computing*, October 2011.

[24] A. Hofleitner, R. Herring, and A. Bayen, "A hydrodynamic theory based statistical model of arterial traffic," *Technical Report UC Berkeley, UCB-ITS-CWP-2011-2, http://www.eecs.berkeley.edu/~aude/papers/traffic_distributions.pdf*, January 2011.

[25] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, Madeira, Portugal, September 2010, pp. 929–936.